Antti Hartikainen

APPLICATION REQUEST AND APPROVAL IN ENTERPRISE

Bachelor's Thesis

Information Technology

February 2016

KYAMK
University of Applied Sciences

Tekijä	**Tutkinto**	**Aika**
Antti Hartikainen	Insinööri	Helmikuu 2016
Opinnäytetyön nimi Application request and approval in enterprise		59 sivua
Toimeksiantaja Salainen		
Ohjaaja Yliopettaja Martti Kettunen		

Tiivistelmä

Yrityksissä on käytössä suuri määrä sovelluksia erilaisiin käyttötarkoituksiin. Niitä tarvitaan yrityksen päivittäisessä toiminnassa. Sovellukset muodostavat monimutkaisen ekosysteemin, jossa on paljon sisäisiä riippuvuuksia. Sovellusten kasvava määrä, monimutkaisuus ja useat yhtä-aikaa käytössä olevat alustat asettavat suuria haasteita sovellusten jakeluun, asentamiseen ja lisenssien seurantaan liittyen. Modernille jakelukanavalle, sovelluspyyntöjen hyväksymisprosessille ja tehokkaalle lisenssien seurannalle on olemassa kasvava tarve.

Tässä opinnäytetyössä keskitytään sovellusten jakelun ja hyväksymisprosessin tutkimiseen sekä kehittämiseen suuressa, kansainvälisessä tuotantotekniikkaa kehittävässä ja valmistavassa yrityksessä. Tarkoituksena on kartoittaa nykyistä tilannetta, siihen liittyviä ongelmia ja yrityksen tarpeita asian suhteen. Yrityksen olemassa olevaan it-arkkitehtuuriin tutustuttiin samassa yhteydessä. Opinnäytetyön tavoitteena on etsiä ja kartoittaa erilaisia ratkaisumalleja, sekä laatia ehdotus siitä miten sovellusten jakelua voitaisiin tulevaisuudessa hoitaa yrityksen sisällä.

Erityisen tarkasti keskitytään sovellusten jakelukanavan ja lisensoituja sovelluksia koskevien sovelluspyyntöjen tutkimiseen sekä kehittämiseen. Tavoitteena on löytää ratkaisu joka tarjoaisi modernin, sovelluskauppa tyyppisen jakelukanavan sekä mahdollisimman pitkälle automatisoidun ja läpinäkyvän sovelluspyyntöjen hyväksymisprosessin. Opinnäytetyössä sivutaan myös asian yhteydessä esiin noussutta ihmisten tiedon puutetta liittyen sovellusten aiheuttamiin kustannuksiin ja sen aiheuttamia haasteita. Tässä yhteydessä huomattiin etteivät ongelmat ole ainoastaan teknisiä vaan mukana on myös inhimillinen ulottuvuus.

Lopputuloksena syntyi useita ratkaisumalleja ja konkreettinen ehdotus siitä miten sovellusten jakelu ja hyväksymisprosessi tulisi toteuttaa tulevaisuudessa. Ehdotettu ratkaisu vastaa hyvin asetettuihin tavoitteisiin. Opinnäytetyön lopputulosta voidaan hyödyntää tulevaisuudessa kun yritykseen aletaan rakentamaan uutta sovellusten jakelukanavaa ja hyväksymisprosessia.

Avainsanat
application, approval, request, application management, Microsoft SCCM

Author	Degree	Time
Antti Hartikainen	Bachelor of Engineering	February 2016
Thesis Title		59 pages
Application Request and Approval in Enterprise		
Commissioned by		
Secret		
Supervisor		
Martti Kettunen, Principal Lecturer		

Abstract

In a modern enterprise there is a large number of applications in use for a variety of purposes. They are needed in daily business. Applications form a complex ecosystem that has a lot of interdependencies. A growing number of applications, increasing complexity and several different platforms used in tandem set a big challenge towards the distribution of applications, installation and license tracking. There is a growing need for a modern distribution channel, an approval process and efficient license tracking.

This thesis focuses on the research and development of application distribution and approval in a large, international industrial engineering and manufacturing company. The aim is to research current situation, possible issues and requirements for the future. The existing IT architecture was also explored. The goal of the thesis is to search for different solutions and propose application distribution solutions for the future.

The thesis focuses especially to on developing the application distribution channel and the approval process for licensed applications. The goal is to find a solution that would provide a modern, application store like distribution channel and automated, see-through approval process. The thesis also takes into consideration the lack of knowledge about application costs that was brought to attention during the making of it. It was noticed that this issue is not only technical, but there is also a human side to it.

As a result, several solutions emerged and a concrete proposal for future application distribution channel and approval process was made. The proposed solution answers well to demands that were set. The proposal can be utilized in the future when new application distribution channel and approval process will be built.

Keywords
application, approval, request, application management, Microsoft SCCM

CONTENT

TERMS AND ABBREVIATIONS

1E	Software company headquartered in United Kingdom developing and selling software management products to enterprises.
Active Directory	Microsoft developed directory service for authenticating users and computers, enforcing policies and software installation.
Application	Computer software or program designed to perform tasks and functions. A web browser for example is an application.
Application approval	Decision resulted by application request. Can be positive or negative.
Application Deployment	A process of making an application available for use. Installation of an application to a computer.
Application Management	A process of managing application lifecycle from start to finish.
Application Owner	A role assigned to a person. Contains responsibilities related to managing application product in question.
Application Request	A request made by user when an application is needed.
Application Store	Digital platform for distribution of applications to end users. For example iOS App Store and Google Play.
BYOD	Bring your own device. Corporate policy that allows employees to bring personally owned devices (phone, laptop) to work and use them access company environment.

License	Permission to use computer software as agreed between customer and vendor.
Line Manager	Head of department or unit in company.
Orchestrator	System Center automation platform for automating different tasks to achieve maximum efficiency.
Powershell	Scripting language and command-line shell for performing and automating administrative tasks in Microsoft Windows environments.
Runbook	Set of instructions which form an automated task or process. Used in Orchestrator.
System Center	Microsoft System Center is suite of products designed to manage enterprise it infrastructure.
SCCM	Microsoft System Center Configuration Manager. Computer systems management software designed to manage large amount of computers.
ServiceNow	A cloud based service management software and platform.
SharePoint	Microsoft developed web application platform. Combines several separate functions like intranet, extranet and document management.
Shopping	Enterprise application store developed and sold by 1E.
Software Center	System Center application for distributing applications to end users. Includes application catalog for browsing application selection.

1 INTRODUCTION

This thesis concerns application requests and approval process in a large, international enterprise (referred as the client company). The aim is to provide a solution to issues in application requests and approval process defined by the client company. The aim is achieved by studying the current situation and reasons behind it, by openly searching for possible solutions and going through the impact they would have, and studying technical aspects related to implementing them. As a result, a proposition about how the defined issues could be solved and how application distribution should be done in future will be made. The proposition should be something that can realistically be implemented both in technical and financial sense. Since the subject at hand is large and complex, it has been decided that the thesis will be tightly limited to application distribution, requests and approvals. Matters like managing licenses and exact technical details will not be discussed.

The client company is an industrial engineering and manufacturing company operating globally in all continents. It develops and manufactures solutions for chemical industry, rotating equipment and pumps equipment. The turnover is around 3 billion euro per year and the number of employees is around 15 000 personnel. This thesis contains information about computer systems and configurations used in the client company, and therefore revealing too much information about it seen as potential security threat. Due to these security concerns, the name of the company and any further information will not be given in this thesis and the matter will be discussed using general terms.

The idea for the thesis was born during summer 2015 when I worked as a summer trainee in the client company. This employment has continued through winter and I have gotten some hands-on experience with application distribution by processing application requests and by several discussions with company employees about current situation. The subject is interesting and very topical since there are many big trends that have effect on how applications will be distributed in future. Since the thesis is made for an international organization, it is written in English.

1.1 World of applications

In today's world, computers are used everywhere and for everything. We live in a digital age where meetings are no longer held face to face and no human input is always needed to get work done. Documents are no longer sent out physically via post and we all carry laptops or hybrid devices with us instead of briefcases containing large amounts of documents. Internet is everywhere, it is not only on your desk in form of a pc, but also in your pocket and wrist. Soon it might be even in your fridge. Most people are reachable vie Internet all the time, no matter the time or place.

A few years ago a computer was a big and bulky device on a desk, or more likely under it. In order to operate it, display, keyboard and mouse were needed. Today the concept of the computer is lot more diverse (Wikipedia 2016a). Mobile revolution has changed the landscape considerably and changed understanding about what computer actually is. Basically, a computer is not only a pc, but also the smartphone in your pocket, the tablet under your arm and the smart watch in your wrist. In future a computer can also be a fridge or oven for example. Modern cars are one of the largest computers out there operated by common people every day. Computers are everywhere, even in places that people do not realize.

Computer runs an operating system (Wikipedia 2016b). Operating system runs applications or programs as they sometimes are called. Microsoft Word, Google Chrome, Adobe Reader, Apple iTunes etc. are all applications (Wikipedia 2016c). They are the component in computers that enable us to get our work done. In other words, a computer without applications would be useless. An average computer has hundreds of applications installed. Usually only a handful of them are actively used, the rest are there only in reserve or because the developer of the operating system has decided to include them in it.

Users can install more applications from various sources. Before the internet, new applications were usually bought from stores and installed from disks (cd, dvd). Applications also usually stayed as they were, since no update method existed. When internet started to reach households, things changed radically. Huge amount of applications for every imaginable purpose was suddenly available for everyone, and installing them was quick and easy. This resulted

in an explosion of the number of applications installed in people's computers. One big change was that a lot of the software in Internet was free. It was no longer necessary to pay retailers to get applications, instead application could be downloaded straight from the source (developer) for free. Internet also made piracy much easier than before. People have become accustomed to free software. Internet also made possible to update applications.

Mobile revolution started the next phase in application distribution to end users. Application stores offered one central location for browsing, discovering, installing and updating applications (Wikipedia 2016d). It is no longer necessary to search the internet for applications, instead just access the application store and install. An application store is very secure and easy to use environment when compared with the thousands of download sites in Internet, which include large amount of unreliable players distributing pirated or unsecure software. The idea about one central location for managing application is not new, many Linux distributions have had a similar system in place for long time. However, it was the mobile platforms that brought the concept to general use. Today most operating systems have some kind of application store in place.

1.2 Enterprise and application management

Applications are important part of daily operations on any enterprise these days. Various applications are needed in order to get work done. In large enterprises consisting of thousands of employees doing various different tasks, a large number of different applications are in use every day. All computers are connected to each other via networks, data is changed between applications all the time. If applications would cease to operate, so would the whole enterprise. Applications form a complex ecosystem that serves critical business functions. It is important to realize that this ecosystem is not formed by just independent applications working on their own, but, instead, different applications share complex interdependencies between them. They are integrated at the platform level. In time, the number of applications in use and even number of different versions of the same application in use simultaneously tends to rise. It is easy to add new

applications but difficult to decommission old ones still in use. A constant need for patching and adding new features results in more complexity.

Application management is an enterprise-wide process of managing operation, maintenance, versioning, patching and upgrading of an application throughout its lifecycle (Wikipedia 2016f). Every application is managed from “birth to grave”. At its core, application management is about the knowledge and control of applications in use, decommissioned from use and to be taken into use in future. It is about keeping tabs on what we currently have on our computers, how we keep it up to date, functional and performing well. It is also about planning the future and going forward, how we respond to the changing business needs in the best possible way.

1.3 Application lifecycle

Application lifecycle is considered here from the client’s perspective. Since the client in question is not a software/IT company, but a company in more traditional development and manufacturing business, I will not take application development etc. into consideration. Lifecycle is presented in Figure 1. Application lifecycle consists of following stages (Advisera 2012).

- The need or idea. “We need to do the following and for that we need some application”.
- Purchase. Potential application is tested (trial-version) and if found suitable, purchase is made.
- Build. Application goes to packaging and further testing.
- Deployment. Application is released to production.
- Maintenance. Application is in active use and it is being kept up to date and functional. Support is provided. Most of the application lifecycle is in this stage.
- Decommission. Application reaches the end of its life and is decommissioned. New installations are no longer possible and no support is provided. Usually, other application takes its place or the need no longer exist.

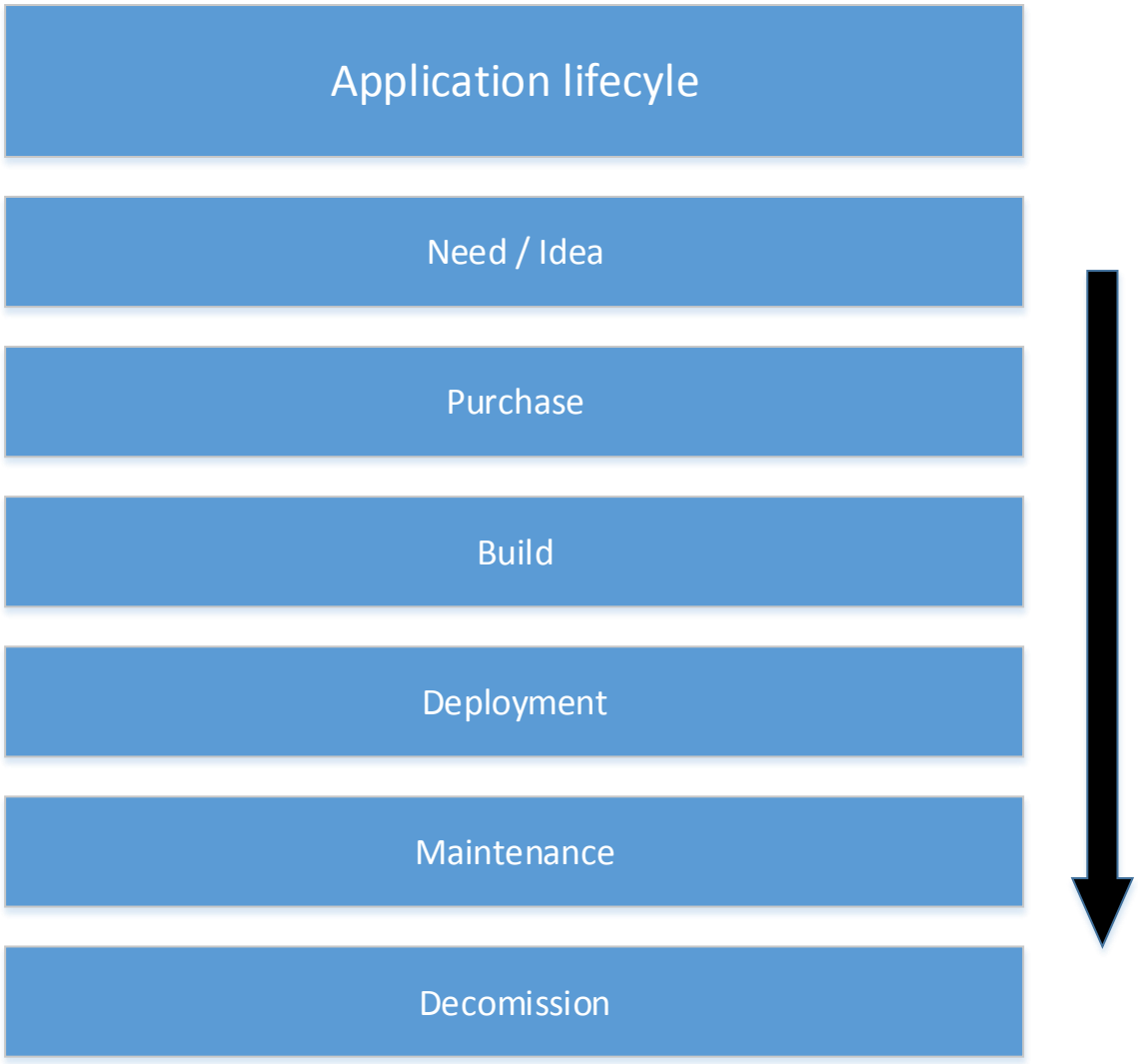

Figure 1. Application lifecycle from birth to grave

1.4 Distributing applications

Before application can be used, it needs to be distributed to end user. There are a lot of ways to distribute applications. However, in enterprise environment it is usually done from one central location which can be easily managed. Basically, a user in need of application can search and install it using a “kiosk” build for this. The kiosk is application itself designed to give user access to enterprise application catalog. It features a simple graphical user interface which allows user to browse and search application catalog. When a suitable application is found, it can, depending on the application, be either installed right away or requested for. This concept is very similar to application stores common in mobile operating systems. These operating systems also feature

one central location (=application) for searching and installing applications. This concept has proven popular and easy to understand and use. Installing applications is usually limited to the catalog found in store, which makes it easy to control what is installed on company computers and what is not. IT can for example check what is installed on an individual computer and comprehensive statistics can be made based on this data. Patches and updates can also be distributed via this system.

1.5 Bring your own device

Bring your own device or BYOD is corporate policy that permits employees to bring and use their personal devices at their workplace (Wikipedia 2015b). These devices can be used to access company environment, data and applications. BYOD devices are usually smartphones or tablets, but also laptops are becoming more common. In recent years, BYOD has been growing in rapid pace and in some markets the majority of corporate employees are using their own devices at work. BYOD is also coming to our client company in the future although no exact timetable has been decided, nor which services will be available.

From the application distribution point of view, BYOD brings some considerable challenges. Until now every user has only had one device, but with BYOD it will be possible that same user has multiple devices, a laptop and tablet for example or two laptops. This can result in situation where user want same software in all devices which in return is difficult because of different types of licenses. The issue is that there two basic types of permanent licenses (not counting floating licenses). One type is one license per user, which does not necessary cause issues since it is not based on devices. User, who has the license, can use it to install software on all his/hers devices. Usually there is a limit for the number of devices, but in most cases, it so high (5 devices) that it will not be breached. Other type license is not based on the user, but the device. In short, one license means that application can be installed on one device. If you have another device needing the same application, second license will need to be purchased. This means that there will need to some sort of tracking and policy in place to ensure that no illegal

installation of software is being made and the license database is kept up to date.

1.6 Application requests

As described earlier, an application can be installed instantly or in some cases they need to be requested for. This is because a large portion of applications are commercial products and therefore not free. Using them requires a license which in return costs money. Since many people have a habit of installing many applications even if they do not really need them, some form of control needs to be applied in order to keep license costs at reasonable level. An average user does not necessarily understand how expensive applications can be. Distributing applications freely without control would result in very high license costs with no real advantage to business.

In practice application request is a notification from user saying that “I need this application for this purpose, can I please have it”. This notification is relayed to the personnel responsible for application management, and they will decide based on defined criteria what is done to the request. Information given by user is very important here. In practice, the request either can be approved, denied or more information can be asked for. When the decision is made, the user is notified about it. If the request is approved, the user can install the application from the kiosk. This process is described in figure 2 where different stages and outcomes can be seen.

Application requests have their own set of issues. Most common problems are related to users not giving enough information about why they need the requested application. Since this is what decision making is based on, it is important to give proper justification for requesting application. Inadequate information leads either to denying the request or in most cases, asking for more information from user. This leads to the exchange of messages between people processing the request and user, which can even take several day when parties are in different time zones. This could be avoided just by making sure that the original request has enough information.

Application request also tend to make other possible issues very apparent. If for example data about licenses, user’s department, position or devices is

inadequate, decision making becomes harder or impossible. Successful decision making requires detailed, up to date data to support it.

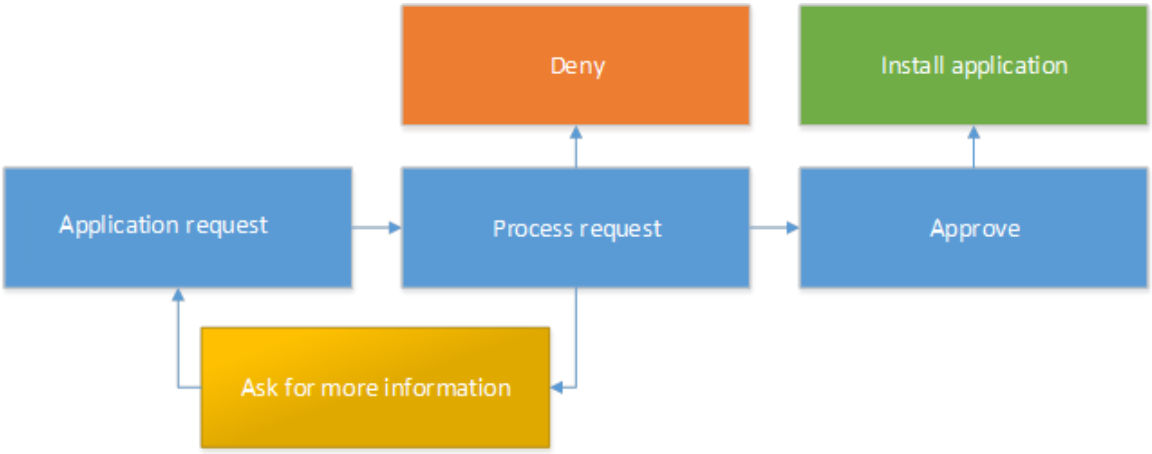

Figure 2. Different stages of application approval process

1.7 User's perspective

From average users perspective application management is essential in order for them to work efficiently, but in a lot of cases it can seem like a nuisance. Typical user is change resistant, when he/she has learned to use certain application efficiently, he/she does not usually like when changes are made. When software is being updated or some cases changed completely, resistance usually arises. Sometimes this even leads to situation where installing updates is avoided for example by asking for an exception, even though the only real reason for this is reluctance to learn new ways of working. In reality there is no option where application could be "frozen" and the same version would be used from now to eternity. Security reasons and ever changing environment alone make it impossibility. In the world of IT change is inevitable and people need to able to adapt even if it is sometimes difficult.

One typical feedback from users is "why can't I just install this right away". This reflects the common lack of understanding about the cost of software. People have trouble of realizing that even if something cannot be physically touched, it can be very complex and expensive. This is especially true in today's world where consumer application are usually "free" in a sense that you do not need to pay actual money to use them. These applications are

funded in other ways like selling the user data for example. However, in enterprise environment applications are still bought.

Traditionally, software distribution has been handled by local the IT people in each department without proper enterprise wide control. Basically, each department was “independent” even if they belonged to the same company. Many people think that it was lot easier just go to the next office and tell what you want, and then have the local IT people install the software for you. Going from this to a global self-service system can be a jarring change and for many it feels like degradation of service.

1.8 Licensing

Licensing is one of the many challenges related to application management. Basically, all commercial software is licensed software. When a piece of software is purchased for use, it is not the software itself that is purchased, but a license, which is a permit to use the software in a way that is defined in license agreement between user and the software vendor (Wikipedia 2016e). In most cases license agreement defines the number of installations you can make. If more installations are needed, more licenses need to be purchased. Installing software without available license is illegal. The license agreements themselves are usually long and complex documents that require expertise to be understood.

In large multinational enterprise license management is a large and complex topic of its own. The sheer number of different licenses alone is a challenge, but when you factor in all possible variables like local / global licenses, begin and end dates, different versions / licenses in use simultaneously, agreements made by different people in different countries in different languages, inadequate documentation etc. you start to realize how big topic this really is. In practice one of the largest issues is simply keeping records about used and available licenses, and making that data available to right personnel. Basically, you need have one party that will be responsible for certain application(s) or one centralized location where all are handled. It is important to make sure that no applications are used without available license. The illegal use of software can lead large costs afterwards.

In ideal situation there would be one global license database that would house all licenses and all relevant data about them. This database could be accessed by people processing application requests in order to check the number of available licenses. The interface could for example be browser based and a wide set of tools for searching and managing licenses would be available. Unfortunately, this kind of license database does not currently exist in the client company and keeping records of available and used licenses has been the sole responsibility of the application owner. This has led to a highly variable ways of license book keeping.

2 MICROSOFT SYSTEM CENTER

Microsoft System Center is a line of products designed for enterprise IT administration (Wikipedia 2016g.). The core function is to help IT administrators to manage a large network of client and server systems including desktop and mobile clients. Microsoft describes System Center as “end-to-end service-management product” that comprehensively assesses, deploys and updates servers, client computers and devices across physical, virtual, distributed and mobile environments. System Center products can be purchased separately or as one complete package.

The core products of System Center are (Microsoft Corporation 2016a.):

- Operations Manager
- Configuration Manager
- Virtual Machine Manager
- Orchestrator
- Data Protection Manager
- Service Manager
- App Controller
- Endpoint Protection

This thesis will concentrate on using Configuration Manager and Orchestrator in application distribution. Other System Center components are not in the scope.

2.1 Configuration Manager

Microsoft System Center Configuration Manager 2012 (SCCM) is systems-management software designed for managing large amounts of desktops, laptops, tablets, smartphones and servers (Wikipedia 2016h). Supported operating systems are Windows, OS X, Linux, Unix, Windows Phone, Android, iOS and Symbian. The core features of SCCM are software distribution, software and hardware inventory, patch management, operating system deployment and client remote control. Using SCCM it is possible to build custom operating system images, software packages and configurations. These can be distributed across the entire environment using one single administrative interface. The aim is to achieve a consistent, flexible, secure and well performing enterprise IT environment (Tulloch M. 2013).

Inventory management is the single most important feature. Clients connected to network are discovered through Active Directory and software is installed automatically. This includes also the initial installation of operating system and other basic software. Application deployment to clients is centrally managed using SCCM, application can be either installed as self-service via software center, and it can be “pushed” to client with no user input required.

SCCM provides administrators with comprehensive reporting tools for gathering, organizing and presenting information about all configuration manager operations in the entire enterprise. These include the hardware and software inventory, software updates, applications and users. There is a large number of predefined reports available and possibility create your own custom reports that suit your needs. SCCM reports are a great tool for gathering data about number installations of certain application for example making them a vital tool in application management.

2.2 Orchestrator

Orchestrator is a tool for automating systems administration tasks. Orchestrator can be used to automate different processes in enterprise environment like workflow automation using visual runbooks (Microsoft Corporation 2016b). It provides simple graphical environment for building automation and does not require writing powershell code to get things moving.

Orchestrator integrates with other Microsoft products like Exchange and can act as “glue” between different components of System Center suite. It is especially good at extracting data from one part of System Center and making possible to act on that data on other part. Automation of basic administrative tasks is increasingly important in today’s IT environments since the number of different systems and devices is constantly increasing.

Runbook is a set of instructions which form an automated task or process. Individual steps forming the runbook are called activities. Runbooks are used to run all orchestrator activities, like application requests and installations. Typical runbook is built like follows: read data -> perform action -> publish updated data. Actions are taken when defined conditions are met. In other words: if (condition) = true -> do (action), if (condition) = false -> drop. Due to the clear graphical representations available in the Orchestrator, it is usually quite easy understand how each runbook operates.

2.3 Software Center and Application Catalog

Software Center is a user facing application that displays available (approved) software, installed software and currently ongoing installation status. It also enables user to configure several options like working hours and remote control. Software Center also provides access to Application Catalog which is user’s gateway to browsing available applications, making application requests and beginning application installation. It is basically an application store although very limited in functionality. Currently, these two components work in their own separate interfaces. Software Center is a Windows-application but application catalog is accessed via web-browser.

2.4 SCCM 2016

SCCM 2016 is the next major release of SCCM and all related components. It is currently available as Technical Preview (“beta”) and final release date is not known at the time of writing this. SCCM 2016 has many new features and improvements, like full Windows 10 and Windows Server 2016 support, Azure VM support and mobile device management (Microsoft Corporation 2016c).

From the application distribution point of view, changes are unfortunately rather small. Software Center and application catalog will be unified to one application which will replace the two separate interfaces currently used. This new application will have completely new interface and some new features. Unfortunately no other remarkable changes or new features will be available.

3 CURRENT SITUATION AND KEY PERSONNEL

In order to understand application distribution, few key terms need to be clear as there are several parties involved.

3.1 Application Owner

Application Owner is the legal owner of application. Since it is not possible for IT department to be an expert in every single piece software in use, application owner is nominated for every application. Usually, the original requestor of application is named the owner of that application. Most owners are not people working in IT, but instead they come from various backgrounds and positions. A business analyst for example can be an application owner. Usually, ownership is an additional responsibility instead of a discrete role (Mac Neela, 2013). The owner's responsibility is to keep tabs on the number of installations and number of available licenses. When a request for licensed application is made, the application owner will check the number of available licenses and based on that either approves or denies the request. If all licenses are exhausted, the owner is to notify procurement about the situation and prompt them to start process for getting more licenses. Additionally, it is the owner's responsibility to be aware of everything related to application in question. The application owner is a contact person and expert. He / She has the best knowledge about the application. In many cases, this requires real expertise, especially when talking about more complex applications like CAD-software.

3.2 Line Manager

The line manager is usually the head of department who has financial responsibility about applications used in his/hers department. Since each

department pays for licenses it uses, the approval of line manager is needed when a licensed application is requested. Basically, line manager is key holder of the safe. It is also line manager's responsibility to find out if the requestor really has a need for the application he/she requested.

3.3 Application Management Situation

Currently there are nearly 200 applications in production. The number of actual different application is slightly smaller since this number includes some different versions of the same applications. On the other hand number of software packages which include all actual applications and all "other packages" is closer to 1000.

At the time of writing this (September 2015), there is no automated application approval process in place. Everything is done manually. SCCM R2 update, which was applied in May 2015, was a notable turning point which ignited the process of rethinking application management.

3.4 Before SCCM R2 Update

Application requests are made via Application Catalog. All licensed applications need to be requested for and approval from line Manager was needed. When user requested for application, an automated notification was sent to the requestors manager (manager is defined in Active Directory). Manager's job was to check if there were free licenses available and if the requestor actually needed the application or not. If both conditions were met, manager approved the request and user was able to install the application. If there were no licenses available and / or request was deemed as unfounded, request was denied.

This model had many severe issues. The basic problem was that the manager had no way of knowing if there were licenses available, there was no tool implemented for sharing that information to managers. Also many of the managers were not that interested in managing application licenses. There was lack of understanding about financial costs that are related to buying licenses and since everything was funded from global IT budget, many

managers think that this is not their concern. In practice this led to a wild and uncontrolled situation where manager basically acted as “approval automatons” that approved every application request that was made. There were exceptions, of course, but the big picture was that usually almost anyone was able to install any application without anyone actually knowing were there enough licenses and was there a real, concrete need for all installed applications. This also led to situation where users always installed the most expensive versions of Microsoft Office Software for example, even if cheaper versions were sufficient.

The result of all this was very high licensing costs as company had to pay for every installation regardless if it was actually used or not, and on the other hand lot of software was distributed even when there actually was not enough licenses. This is the situation that cannot and will not come back.

3.5 SCCM R2 Update

In May 2015 SCCM R2 update was applied resulting in a new situation. Components previously used in application requests were left unsupported. This was decision made by Microsoft and it resulted in a technically broken application approval process. This was actually a known risk which was taken because there were other reasons for installing the R2 update. Skipping the update was not an option. After the update was applied, automated notifications about application request were no longer working and it was decided that the whole process would undergo rethinking.

Since a new process was not available, as temporary solution all application requests were handled manually. This changed the process considerably. When user applied for application, the request appeared in SCCM queue and waited for someone to take it in for processing. The queue was checked for new requests twice a day. This triggers the approval process which consists of many stages. First it is checked that who is the application owner of requested application and an email is sent to him/her asking whether the request can be approved or not. The application owner then decides what is to be done with the request. If the request is for example about more expensive version of Microsoft Office Software (i.e. Visio Premium) and no valid reason for this is

provided, the request is “challenged”. This basically means that user is contacted and reasons for the request are demanded. There is no automation of any kind at any stage of this process, but it is full manual labor.

This process has many issues:

- There is no line manager associated in the approval of requests.
- Application owners have no way of checking the amount of available licenses if they do not have their own manual bookkeeping of them. Usually, this leads to situation where owners approve requests even if they do not know if there are enough licenses or not.
- Client-team needs to take part in the process because there is no automated notification to line managers and application owners about new requests.
- Client-team, user and application owner can be in different time-zones leading to delay in process. Handling a request can take several day because of this.

When starting this thesis, the process described is still being used.

3.6 Line of Management

Company management has outlined some basic principles how application management should be done.

- Self-service should be used as much as possible. The application installation process should be automated as much as possible.
- There is no longer “one large IT budget” covering everything. Application will be charged from the department or company that is using it. If you use it, you pay it.
- Company will stay in the scope of licenses we have. If there no available licenses, no application will be installed. You cannot distribute what you do not own.
- Not all licenses will be global. License can be purchased for use in certain department or country only. This means that software will not be used anywhere else.

- Purchasing new licenses and making agreements will be handled by Procurement and Procurement only.

3.7 Staff opinions

Several engineers working daily with application management and SCCM were interviewed regarding application requests. Since the company line has already been established, they were asked to tell their own personal opinions and views about the matter. The aim was to see if there were any differences in people's views and possibly get new wild ideas.

3.8 Main issues currently

When asked what the main issues currently are, two matters were raised: lack of automation, lack of data and lack of maturity in application ownership. The lack of automation is a technical issue meaning that everything is done manually requiring a third person to take part in the process as described earlier. This needs to be solved by building automation based on SCCM Orchestrator for example. Lack of data refers to the missing up-to-date license database and difficulty of using available data. Lack of maturity in application ownership refers to lack of understanding between application owners about the importance of controlling licenses. Key reasons for this were said to be lack of data and lack of interest about the matter.

"Technical matters are not the biggest issue, people are. Managing licenses is not the main job of people, because of this many of them are not interested in this."

"Previously there was a fully automated process, but it did not work because people were not informed / committed to it. The owner should have had their own book-keeping."

"Earlier automated process was technically good but people doing the approvals had no clue about it. Everything was approved and no licenses were checked"

“Earlier everything was done locally, every unit had their own Windows images and local licenses, everything was done locally. Now we have transferred from local to global, it is governed centrally.”

“Automation has to be applied. This is paramount”.

“Current situation is untenable”.

“License database is needed, a completely new one, not the current procurement crap we have”.

“The old process had no data about cost and licenses. Nothing at all.”

The staff seem to generally agree that technical issues are not the only paint point, getting managers and application owners to commit to the process is a big challenge and might prove to be the most difficult part of implementing the new process.

3.9 Thoughts on the new process

Currently a new process based on SCCM Orchestrator and a two-stage approval is being worked on as described earlier. The staff were asked how they feel about this new process.

“Technically a good model, line manager has to be financial manager (budget owner). The cost needs to be shown“.

“There are many matters to think about, who are taking part in approvals and who are not.”

“This alone is not enough. IT must produce a report for the application owner from which he/she can see the license status. Manager needs to see cost information”

“Chaining is an issue. What do we do when the owner for example is on long vacation”?

Generally the proposed process was seen as a good way of processing application request.

3.10 Showing cost information

During the interviews, matter of costs was constantly brought in attention. It was highlighted that bringing cost information to the attention of managers processing application request is extremely important. Otherwise it is difficult for them to outline what approving a request actually mean in terms of money. Especially many CAD-application have very expensive licenses. It was also noted that in 2016 every department will pay the licenses they use from their own budget. There will no longer be “one large IT budget” covering everything.

“*When presented with the price, a lot of people suddenly do not need the application.”*

“In 2016 local IT billing will really begin. If you use it, you pay for it.”

“Why do we have local licenses? A global license would surely be cheaper and easier. If you use 10 licenses out of 100, you pay 10% of the license cost.”

“Local IT billing will have a considerable effect on the number of applications people have on their computers, it will decrease considerably.”

“Application request needs to have information about license cost”.

3.11 Role of the Application owner

As mentioned earlier, lack of maturity in application ownership is seen as a big issue. It a possible weak point in the new process that might cause issues even if the process is technically successful. If people are no committed, this will not work. Importance of a functional license database and ease of checking license status was constantly highlighted.

“It should be in the owner’s interest that their software works and users are satisfied.”

“Owner is not just a license checker, he is also a “technical approver” who has wide knowledge about his applications. A well-informed owner knows well different local- and global licenses.”

“If the owner is not available or doing his job, perhaps we should have an automatic block for installing the application if licenses are all used.”

“The owner says new version came out, current version does not work etc. Then makes a request to Sharepoint about the new version, it is packaged and owner checks if it works. Released to distribution. This is not proactive, it only happens if there is a problem.”

“There is a lot of software that requires continuous maintenance. The owner is a contact person who is contacted when information is needed for example about new versions and who is going to test it.”

3.12 License checking

One way to handle checking licenses after manager’s approval would be giving that responsibility to IT department. The idea behind this is that then everyone could be sure that license checking would really be done properly. It would be handled by people who are professionals and committed to the matter. This alternative model sparked many kinds of opinions.

“This is how a great number of applications are done right now, general company wide applications. CAD-Applications are separate matter as well as some business critical applications.”

“Approvals can be done not including CAD-applications. “

“Updates are impossible for us to do because there is too much work.”

“In any case new license database is absolutely needed. Otherwise this will take too much time no matter who is doing it.”

“Perhaps we could have one approver per area.”

“Processing requests here is not a good idea since we do not have enough information and know-how about applications. CAD-software is the best example of this.”

“When asked, over 30% of ThinkCell users told us that they do not need it even though they have it installed on their computer. Instead of buying new licenses, we can free a lot of them just by asking.”

“If owner’s role in application approval is just to check licenses, then that could be done automatically. A script will check license from the database and approves or denies based on that.”

3.13 Communication “black hole”

One important matter that was brought to attention during interviews was lack of communication between user and people who process requests. This is largely a result of Software Center lacking any kind of means to change information between people. There is no possibility to assign request processing to people, which means that user has no idea who processes the request, or is it being processed at all. Furthermore, Software Center does not provide any kind of communication platform. As a result:

- User does not know if request has been received / forwarded.
- User does not know who processes the request.
- User does not know what the status of the request is.
- User has no way of asking questions.

Basically, this is a completely opaque process where user sits in dark waiting for any information about his request. This especially problematic when processing request takes a long time and no communication towards user is provided. In these cases this usually results in many emails being sent where user is trying to find out who to contact regarding the request.

3.14 Information needed in application requests

When decision making people receive application requests, they should contain all necessary data needed in decision making. Asking for more information afterwards is time-consuming, can cause delays and prevents user from getting work done with the needed application. Some of this information can be added automatically, but the rest has to be added by user when making the request. The most important piece of information is reason for requesting application. User must tell why he/she needs the application and for what it is used for. This reasoning is the single most important factor when requests are approved. It is important that this is communicated to user

clearly so that number of request lacking all necessary information can be reduces to minimum. All required information is described in Figure 3 where it can also be seen what is added automatically and what needs to be filled in by user themselves.

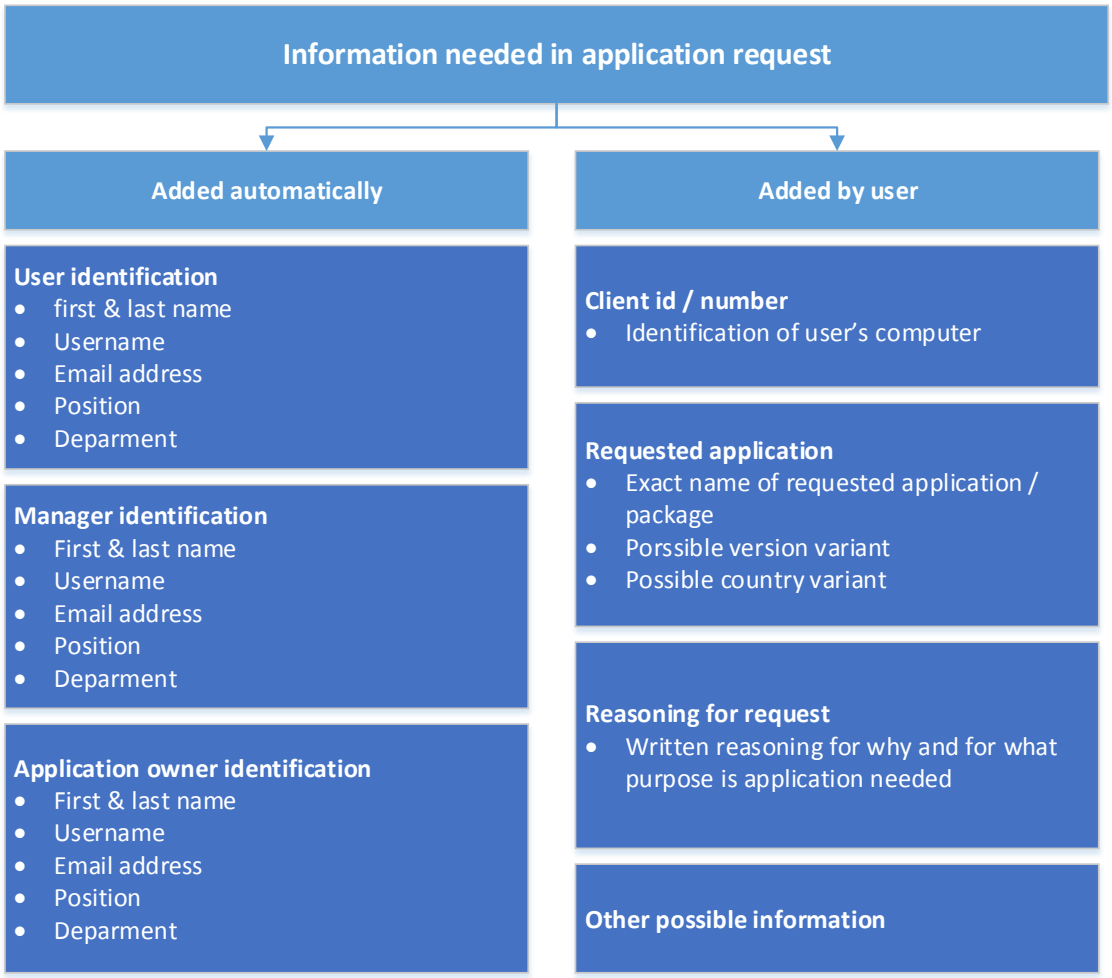

Figure 3. Information needed in application request

3.15 Application catalog of dreams

These days everybody is using smartphones and other mobile devices, which all have application stores. An application store is essentially one central location that provides users with all functionality related to browsing, installing and updating their applications. These application stores are very close to what is needed here: A good-looking, easy-to-use interface, where browsing and searching applications are intuitive and “web-like”. Applications can be divided into categories making discovering them easier and user permissions

can be given based on category. Applications can also be defined so that they are available for example only for users in certain geological location, as there are “local applications” (local license) in use. Each application will have its own homepage which has comprehensive introduction about it including description and screenshots. Description should be accurate, up to date and easy to understand so that number of incorrect request can be reduced to as small as possible. Other important information, such as price and application owner, will also be displayed. Price information is especially important since it will increase general understanding about the cost of software and should reduce unnecessary installations considerably. In practice, it should encourage people to install possible cheaper or freeware variants instead of more expensive applications. The price must be presented so clearly that it is almost impossible for user to miss it.

Application catalog should also offer a see-through approval process where all parties are aware of what is going on, what is the status of request and who are processing it. User must be able to see the status of their request and who are processing it. Suitable interfaces for processing requests should be available to application owners and managers. This should include a summary view where all ongoing activities can be seen at one glance and individual request can be opened for closer inspection. Users should be provided with a easy to understand view of currently installed applications and their history of installations and requests. Figure 4 displays functionality and information that should be available in application catalog in each of the most essential views or pages.

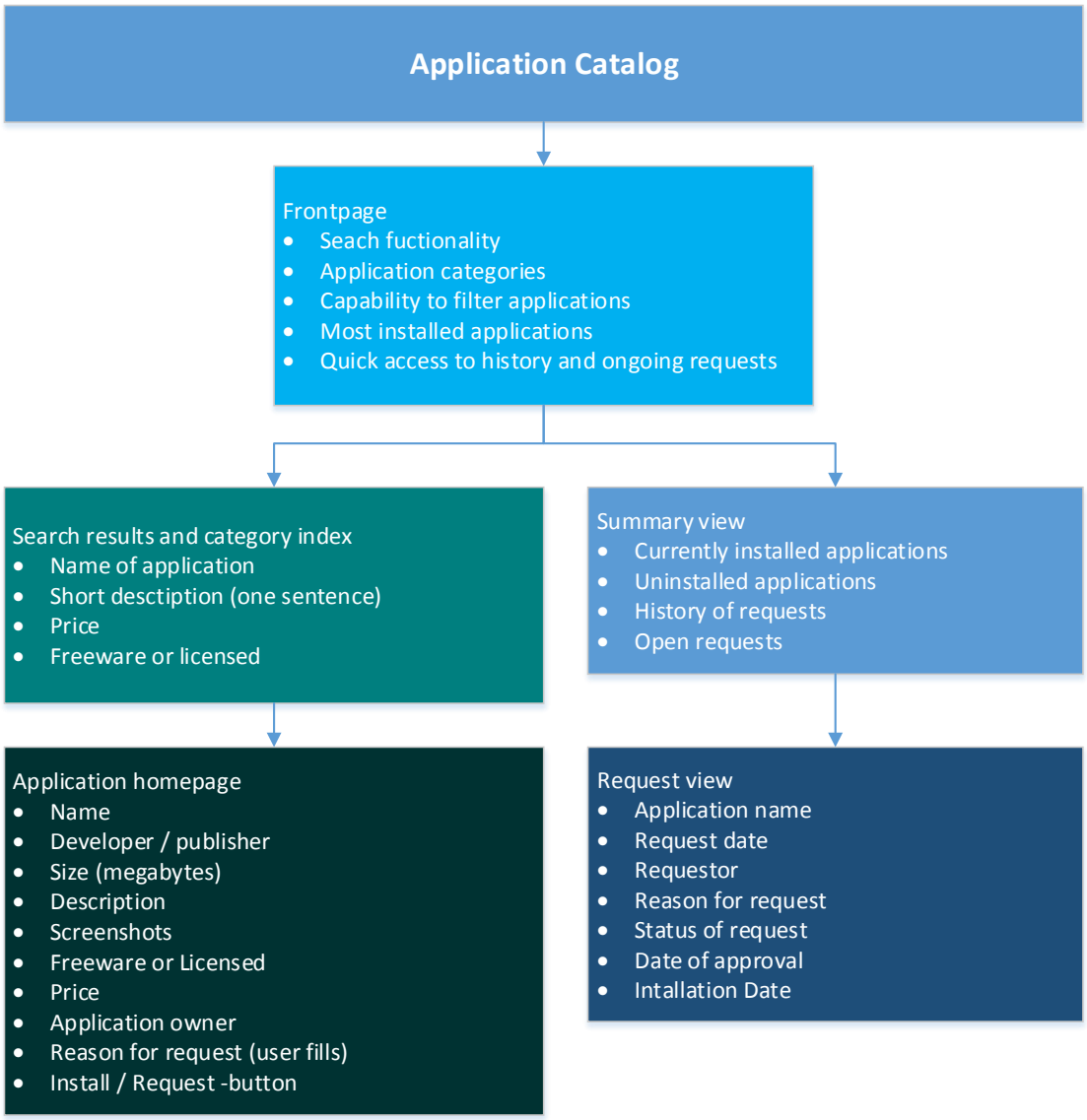

Figure 4. Desired functionality in application catalog

4 THE SOFTWARE CENTER MODEL

The ultimate goal of this thesis is to find new, concrete ways to process application requests in the client company. Basically, this means introducing new processes step-by-step and analyzing their advantages and weaknesses. As a conclusion a recommendation will be made. This model is basically a more advanced version of what was done before the SCCM R2 update. In short, Software Center will be used as the user facing interface and requests will be forwarded to line managers and application owners automatically via email. The email message will either feature an interface for decision making (approve or deny), or will direct user a Sharepoint site where input will be given.

4.1 Processing a request

Software Center will continue as the interface used for browsing, installing and requesting applications. Freeware applications can be installed right away, licensed applications need to be requested. The basic idea here is that when an application request is made, it will be forwarded to user's manager (line manager) and the application owner for approval. User, user's manager and application owner need to be identified in order for the request to reach right people. User identification will be done automatically since each user is logged in to company domain with their individual account. Manager will be identified from the active directory, where every user account has data about manager. The application owner will be identified from SCCM where every application has an owner marked.

Based on these identifications, a notification email will be sent first to line manager, and if he accepts, then to the application owner, who will make the final call. Both parties will need to approve the request in order for user to install the application. If either party deny the request, user will not be able to install application. This process is described step by step in Picture 5. Client-team would not participate in this process in anyway. This would purely be between user, line manager and the application owner.

4.2 Implementation

There are two different ways to implement this. The process can be based entirely on use email messages, or Sharepoint can be utilized as the interface for request processing. In each case notification email needs to be sent to alert parties that they are needed. This email can be more than just simple alert about new request. It can contain the actual interface for request approval, meaning that the email contains information about who is requesting, what is requested, why it is needed and finally a way to approve or deny by simply pressing a button (Peterson 2012). This would be achieved by "approve" and "deny" –buttons included in the email. These buttons are actually links which execute desired action and open a web browser where a message is displayed telling that "Request has been approved / denied". An email message about the request would first be sent user's manager and

based on his decision will either be forwarded to the application owner or will be dropped. In the latter case user will receive email telling that the request has been denied. In case of approval from manager, request will be forwarded to the application owner, who will check if licenses are available and either approves or denies based on license availability. In either case user will be notified via email. If request is approved, user can install the application from Software center (Peterson 2014).

This process could also be done using Sharepoint as the interface for processing requests. Notification email about new request would still be sent, but this email would only contain a link to Sharepoint site, where actual data would be located and where decision would be made and input is given. Similar “approve” and “deny” –button would be located on this page. The basic process will be the same here. The interface is just different.

4.3 Sharepoint or Email

Sharepoint and email have their advantaged and disadvantages. Using only email messages is simpler since it would not involve adding another component to the process. Manager and the application owner will receive one email message, which is not only notification, but the interface for decision making. Read the email, make decision and click either approve or deny –button in the message. No other steps are required. This is very straightforward for the end user, and easier to implement. The disadvantage is that when compared with Sharepoint, email is a much more restrictive format when it comes presenting information.

Using Sharepoint would enable much more refined ways of presenting information, meaning a graphical interface with all information presented in a more modern way combining the use of graphics and text. The disadvantage is that it will also add complexity to the process technically as well as from user perspective.

4.4 Issues

Software Center has several issues as an interface as it is basically impossible to customize. There is no possibility to add custom fields to the application page, which means that no cost information or screenshots for example can be presented to user. It does not allow adding custom fields to application request form, nor is there possibility to show status and people processing it. Adding questions and requesting for more information cannot be done as no communication platform is provided. In short, no approval workflow of any kind is provided, resulting in no way of tracking the process and earlier mentioned black hole in communication between user and people processing requests. The application catalog itself is an inflexible list of applications that does not allow separating applications to categories, which would make browsing and discovering applications easier.

Some of these issues can be circumvented by using Sharepoint as mentioned earlier, but fact remains that as catalog interface Software Center is highly limiting. Requirements presented earlier will not be met here. This is more like “a bare minimum” –model, where the very basic functionality will be achieved, but no real progress towards an application store like experience will be made.

It is interesting that Software Center is so limited in functionality, since all the needed components and data exist in SCCM. All that is missing is an interface for presenting that information to users. This highlights well how much enterprise software is lacking behind of software designed for consumer use.

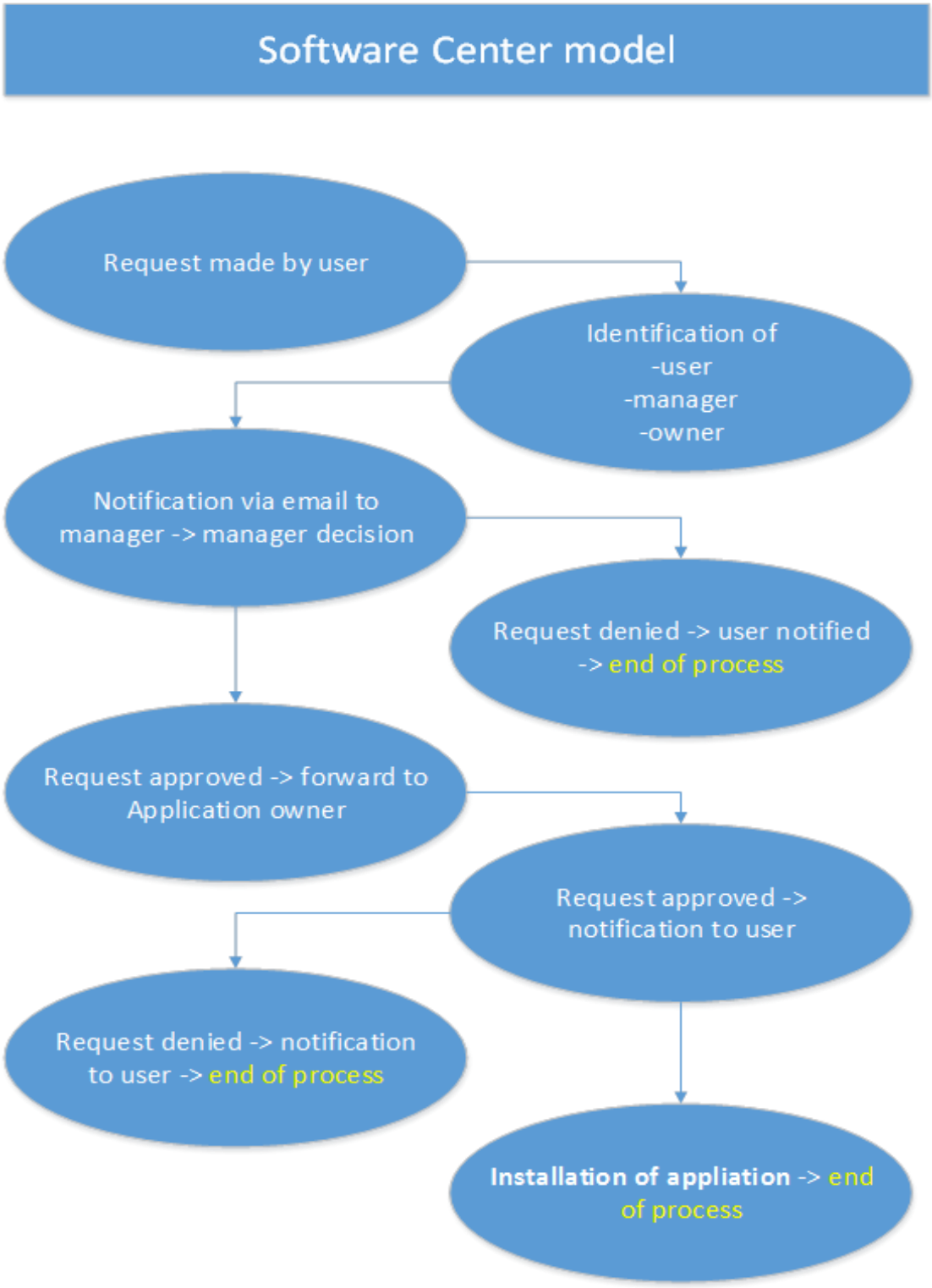

Figure 5. Application approval from start to finish in Software Center model

4.5 SCCM 2016

As mentioned earlier, a new version of SCCM will be released in 2016 (Microsoft Corporation 2016c). From an administrative point of view, it is ideal that all functionality would be available in the core product (SCCM) itself, and

no need to implement additional solutions would exists. Additional solutions tend to increase complexity and add costs. Keeping things as simple as possible is usually a good approach. Although the new SCCM version will bring several important improvements, it seems that Software Center, which is a key component in application deployment, will not gain significant new functionality. Software Center and application catalog will no longer be separated into two different interfaces, but instead one unified application will be used. However, despite the obvious need for proper, customizable application catalog interface and see-through approval process workflow, Microsoft seems to have chosen not to improve these components, and as a result the new Software Center is basically a cosmetic update over the current version. It does not offer solutions of any kind to the severe limitations that the current Software Center has and therefore will not meet the demands set for a new application request process. This is highly disappointing and in practice gives a lot of ground to rival products.

5 SERVICENOW MODEL

During spring 2016 ServiceNow will replace currently used Microsoft SCSM as ticketing System. ServiceNow will be used for It Service Management processes like service requests, change requests, incident management etc. This system provides a method of assign tasks to people, track workflow step-by-step and can act as a communication platform between different parties. All actions will be made from beginning to end using ServiceNow interface, Share Point will no longer be used for makings tickets.

Service Now will be fully administered by Stefanini, an external company, who will in co-operation with clients Service Management Team to ensure the stable operation and proper execution of changes. This means that no administrator rights will be handed over to the client and execution of all changes will go through Stefanini. This is a stark contrast to the current situation where the whole service management process is owned and administered by the client company themselves. Although ServiceNow is not purchased for application deployment in mind, it can be utilized for it.

5.1 What is ServiceNow

ServiceNow is enterprise Service Management software developed and marketed by ServiceNow Inc, which is a platform-as-a-service (PaaS) provider. ServiceNow Inc was founded by Fred Luddy in 2003 and today it is listed in New York Stock Exchange (Wikipedia, 2015a). It has over 800 enterprise customers globally and turnover of around 1 billion US dollars. Headquarters are located in San Diego, California. It is one now of the fastest growing cloud companies in the world.

ServiceNow (software) itself is a cloud software and platform designed to provide customers with a comprehensive workflow / process automation in various fields of business like IT, HR and finance for example. It uses a modern web-based interface which is accessed using a web browser. It is important to notice that ServiceNow is not only a piece of software, but an open platform that enables third party software development and provides an application store which can be used to distribute applications to customers. This model resembles a modern mobile operating system, where you get certain features out of the box, and when more is needed, it can be purchased from the application store. This means that there is lot more available than just the basic functionality built into the core software.

5.2 Application Catalog

The basic idea of the ServiceNow model is that Software Center will no longer be used as an interface for browsing and requesting applications. Instead user will access ServiceNow portal via web browser. The application store will be located here. Unlike in Software Center, this application store is highly customizable and will provide users with similar experience as many well know consumer products. Applications can be divided into different categories to make browsing easier and there is also a search functionality. Each application will have their own page in the Application store containing all necessary information, like the price and name of the application owner for example. Applications can be installed (freeware) or requested (licensed) by clicking a install / request –button.

5.3 Application request

Requesting licensed application will automatically create an application request process which user can see in ServiceNow interface. A request will be automatically assigned to the right people according to the requested application (application owner) and person requesting it (manager). People involved in the process and status of the request will be displayed, meaning that this a transparent process giving user information about what is going on. When new application request is created, the application owner and user's manager will be notified about it via email. They can access the request via their own ServiceNow interface, where they can view and process all requests assigned to them. This single view provides a quick and easy to use glance at everything that is going on. All parties will use the same interface, only difference is that they have different permissions and therefore different features at their disposal.

5.4 Application installation

Even though Software Center is no longer used as application catalog, it will still exist as an installation interface. When user gets permission to install application, it will be automatically pushed to user's computer by SCCM and installation will start in the background using Software Center. No user input is needed. This works the same way as the current system does. Current on-going installation as well as all past installations can be viewed in Software Center. This highlights one important fact here. Even if user's will only see and use ServiceNow when they need new applications, it is only an interface for processing application requests. The actual process and installation is still executed by Microsoft System Center to which ServiceNow integrates into. What is changing versus the current situation, is the user facing interface. Backbone will remain the same, meaning that this is not necessarily as huge change as it first might seem. It is also possible to implement an automated incident reporting. In case of installation failure, a ticket about it can be automatically created. In Software Center application catalog installation failures happen silently.

5.5 Implementation

As mentioned earlier, Service Now will be integrated into existing System Center infrastructure and will act as an interface for application store and application request processing (ServiceNow 2015). The rest will be done by System Center. No third-party orchestration platform is needed and existing SCCM infrastructure will remain unchanged. The configuration Manager contains all applications and their data, as well as application collections. Application data is imported to the application store and then displayed to user. Active Directory can be used to identify users and manage access to applications via group membership if needed. ServiceNow is integrated with Active Directory and syncs user accounts with it. System Center Orchestrator will act as the glue keeping all components together and making sure they will function as needed. Orchestrator runbooks will perform the tasks, which result in a request software item in ServiceNow translated into an installable item on user's computer. These runbooks will monitor Service Now for new requests, commence the approval process, populate SCCM application groups and dynamically configure SCCM to push requested application to user. This functionality is presented in figure 6 where it can be seen how Orchestrator monitors ServiceNow for new requests and acts when one is found. Since all this functionality happens automatically, no Service Now administrator accounts will be needed in order accomplish daily tasks.

Here is another important difference. This model is something that will not be built and implemented by the client themselves. Instead application stores like this are sold as service by third party software developers. In essence, ServiceNow is an open platform that enables third party development and additional features not present in the base product. This is very similar to Active Directory for example, where you can improve functionality by adding applications which provide tools for managing AD objects and exporting or importing data. These services are usually invoiced through the existing Service Now subscription relationship. ServiceNow has its own Service Catalog which can act as an application store, but it only creates requests records that have to processed manually. This is not enough and a better implementation is needed.

Since a working SCCM infrastructure is already in place, it is important that application store integrates fully with it. In practice this means that application store should be automatically discovered and displayed in store without the need for manual work. Approved application requests should result in automatic installation to users or self-service installation via application store.

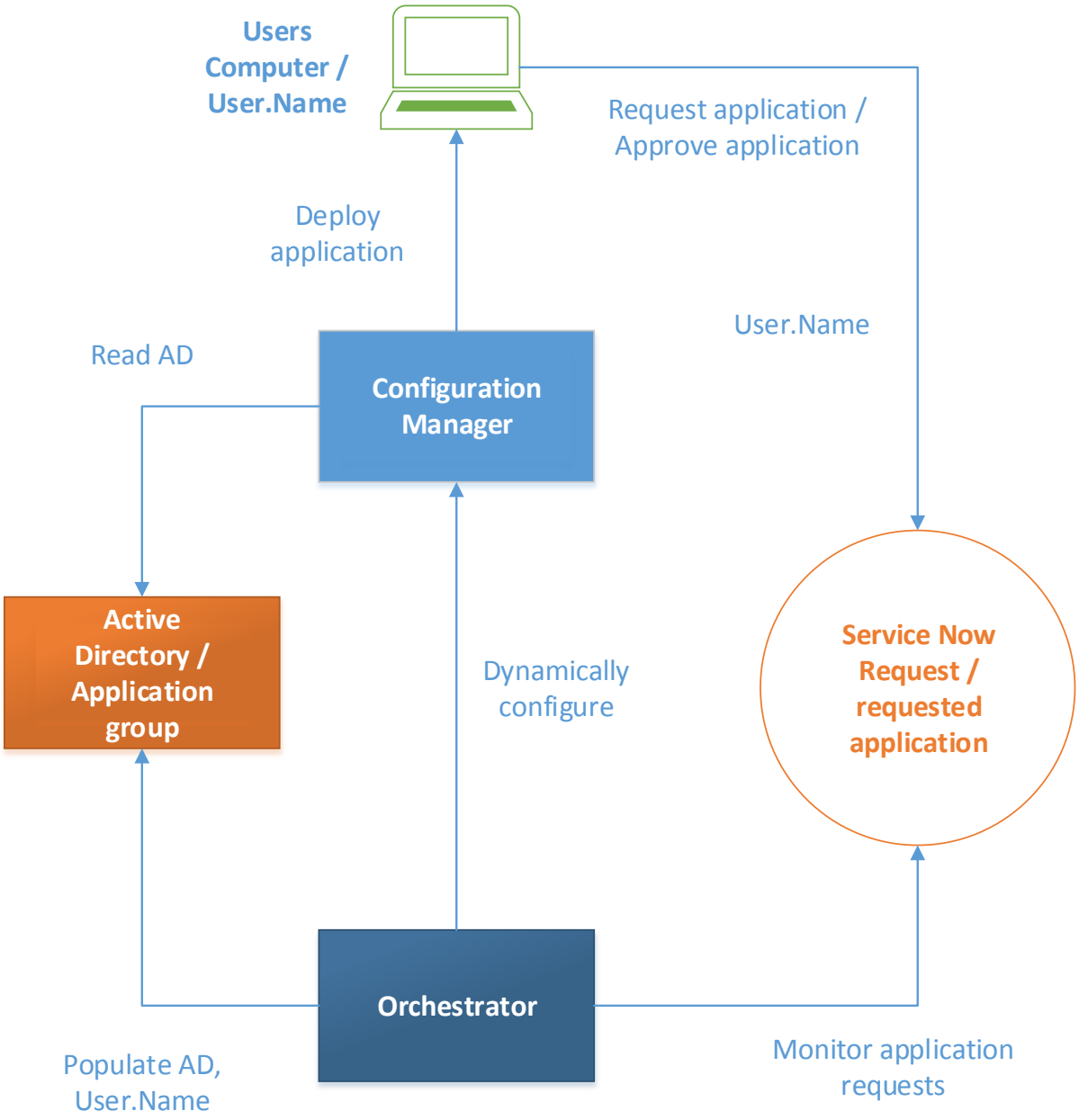

Figure 6. ServiceNow integration with System Center infrastructure

5.6 1E Shopping

1E is a software company focused on developing and selling software lifecycle automation products to enterprise customers. 1E was established in 1997 by ex-Microsoft employees. The company is headquartered in London and has regional offices in Dublin, New York and New Delhi. 1E has over 1700 organizations as a customer in 42 countries. These include several well-known names like Dell, Verizon, Nestle and Ford. The smallest customer has 500 employees, the largest more than half a million. 1E Shopping is an enterprise application store that integrates into ServiceNow and Microsoft SCCM as well as Active Directory architecture (1E Limited, 2015a). It provides corporate users with consumer application store experience very similar to Apple App Store, iTunes and Google Play for example.

Originally Shopping was a standalone product, but today it can be fully integrated into SCCM environment and ServiceNow (1E Limited, 2015b). The shopping application store (from now onwards referred as “store”) is a web application accessed via web-browser. In our case it would be integrated into ServiceNow interface and accessed via ServiceNow portal. Store is highly customizable and feature rich. Basic functionality includes application categories, search functionality, customizable application pages and capability to quickly make application request with a single click. Store look can be customized to fit corporate identity. The store is able to make use of application data found in SCCM, but in addition each application available in the store can be customized with additional data seen only in store. This data can be, for example, the price, description, the application owner and approvers. One of the most important features is workflow for application requests through ServiceNow. When a request is made, an incident is created in ServiceNow. Incident can be automatically assigned to correct people (application data) and notification emails can be sent if wanted. The approval itself can be made via ServiceNow interface using ticketing tools or by email. Approval process has status that is updated as inputs are made. This means that all parties are aware how a request is being processed and who are involved. The application can directly be pushed to user’s computer after it has been approved.

5.7 Shopping application workflow

When user accesses the application store via ServiceNow portal and requests for application, a script is called and a ticket in ServiceNow is opened using parameters passed from the store (Godfrey I, 2015). The ticket will get an ID (GUID) from Shopping and this is used to identify and modify it. When a ticket is opened, some data needs to be pushed from store to the ticketing system. ServiceNow requires the following parameters in order to open a ticket:

- User Name
- Category, Sub Category
- Ticket Number
- User email
- User machine name
- Assigned to, Assignment Group, Opened By
- Impact, Urgency, Priority
- Additional comments

These parameters are stored in an xml file as static data and Shopping pushes these to ServiceNow where parameters are transformed from an import set table to an incident table in order to open a ticket.

A ticket is always created even if approval is not required. If no approval is needed, the ticket is automatically set as approved and installation script is called. Application will then be pushed user's computer by SCCM and if installation is successful, ticket will be marked as closed in ServiceNow. In the case of failed installation, the ticket will not be closed and it will have to be processed manually.

If approval is required, an approval workflow is started after the creation of a ticket. Workflow can build so that approvals are done in chain if there are multiple approvals needed. The ticket can be assigned to user's manager first, then to the application owner etc. Every time the approver gets notified and makes decision, the ticket is updated and the next stage in workflow is initialized. When approval workflow is completed, the ticket is set as approved and installation script is called. Installation will then be started by SCCM and if successful, the ticket is closed. All scripts used are customizable vbscripts. At every stage of this process, all actions, decisions and comments can be seen in the activity section. User can also cancel request when it is pending for

approval. If cancellation is made, a script is called and the ticket is set as closed. Approval workflow is represented in figure 7.

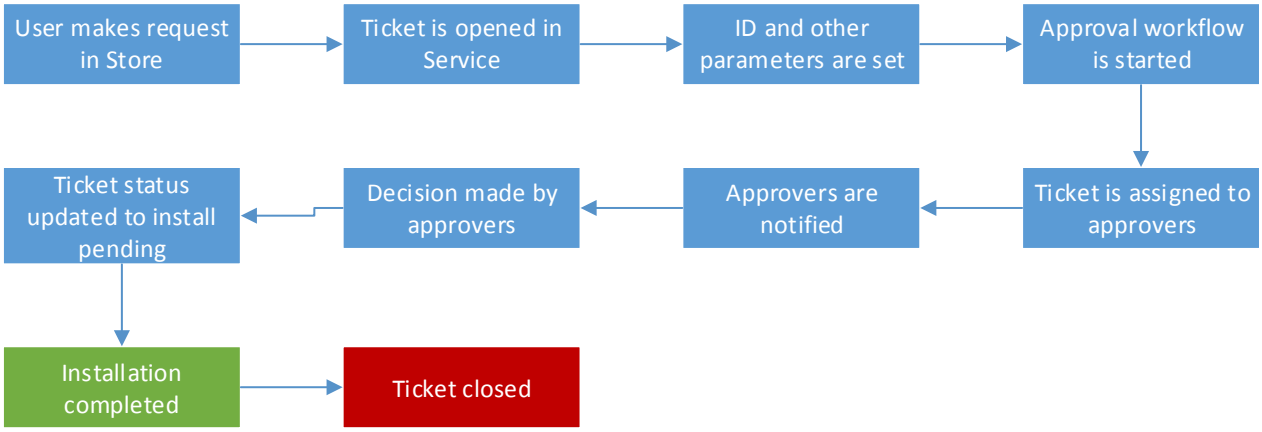

Figure 7. Approval workflow in ServiceNow

There are four vbscripts involved in this process (1E Limited, 2015s).

- ApplicationRequested.vbs
- ApprovalUpdate.vbs
- ApprovalCompeleted.vbs
- InstallationCompleted.vbs

ApplicationRequested.vbs passes necessary parameters from Store to ServiceNow and open a new ticket. ApprovalUpdate.vbs is used to modify tickets status. ApprovalCompleted.vbs modifies ticket status and set installation as pending. InstallationCompleted.vbs indicates that requested is completed, application is successfully installed and closes the ticket. This workflow is presented in figure 8. All the scripts presented here are templates and they can be customized to fit customer's needs. This workflow is a single threaded process that will only execute one script at a time. No following scripts are executed until the previous one is completed and no data is generated in bulk. Also no more than 1 transaction of data from Shopping to ServiceNow is executed per second. Execution of shopping scripts should have no performance effect on ServiceNow.

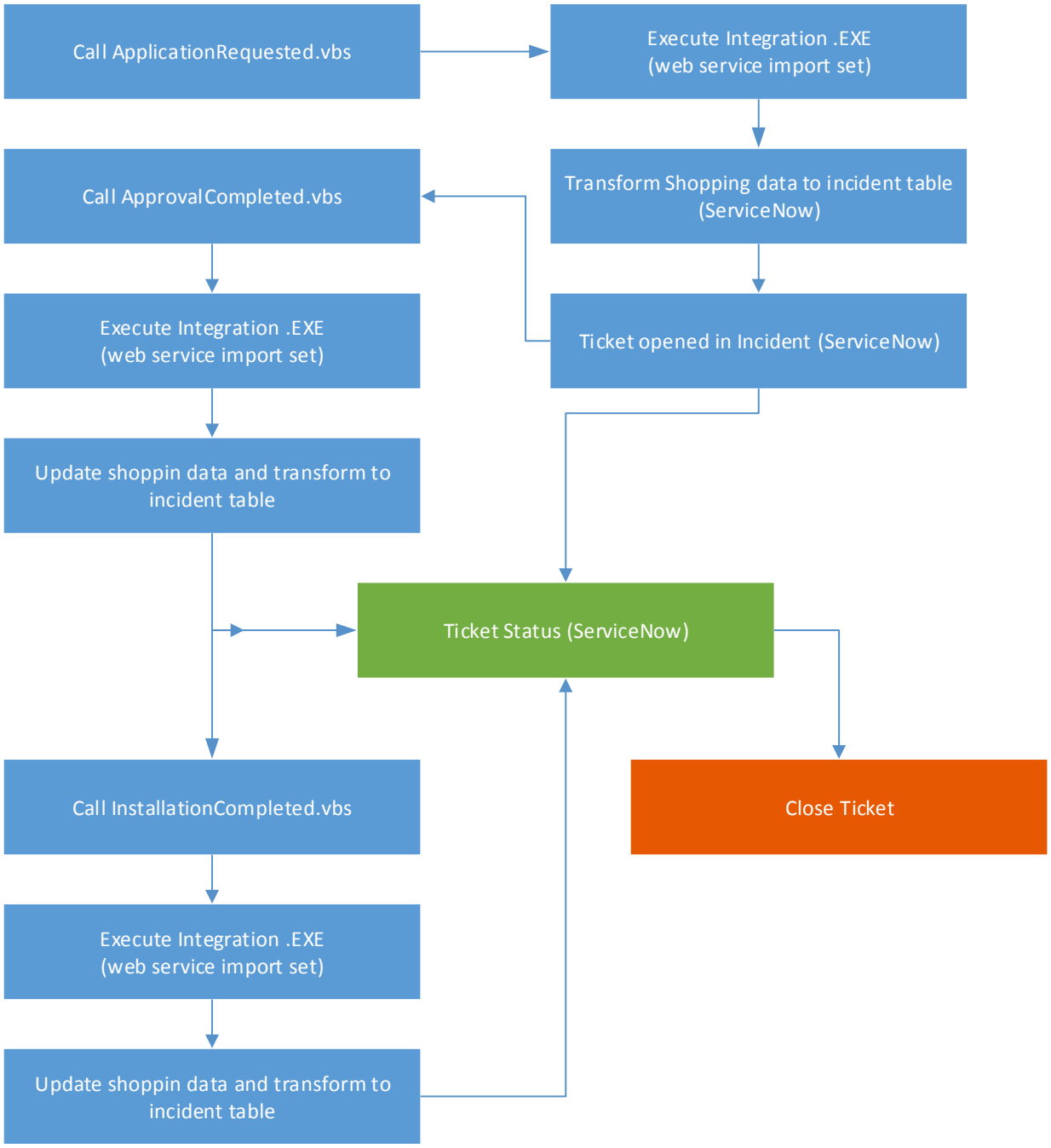

Figure 8. Script execution in application approval workflow

5.8 Advantages

ServiceNow together with the Shopping store has several notable advantages when compared with Software Center. The application store can be customized to fit the client's needs. These include dividing applications in categories for easier browsing and discoverability and versatile application pages which can include for example description, images and price information. The experience resembles app stores used on mobile platforms

like Google Android and Apple iOS. In Software Center you only have one, big static list of application and very bare bones application pages with no possibility for price information.

Application requests can be automatically assigned to right people and Service Now provides a view and tools for tracking the process. User, manager and the application owner all see the current status of the process, what has been done and what is still missing. It is a fully transparent process that does not leave end user into dark about what is going on with his request. This eliminates the need to manually assign requests to people as well as need for sending emails asking for approval or more information. People processing request will also have a proper interface for it, instead of email solutions. There is a huge advantage here over Software Center, where no tools or view for tracking the process or assigning it automatically to people is provided.

Finally, using ServiceNow and Shopping as an application store means that there is one interface and place for IT support and application requests. This is easier for user who does not need to learn how to use two different systems. Simpler is better.

5.9 Disadvantages

Because ServiceNow is being bought as a service from external company, who is handling all administrative tasks, there are some questions about how this will affect managing the application store and approval process. Since no administrative rights will be provided outside the service provider, all changes will have to go through them. This means that there might be extra steps and complexity in daily administrative processes, which increases the time needed to complete them and adds the possibility of errors and misunderstandings. There are also additional costs since ServiceNow and Shopping are commercial products. Using just Software Center would not involve extra costs and would maintain a clean Microsoft environment. There are also some question marks concerning integration to existing SCCM architecture. When software package is updated for example. How will it be updated into the store? Will a manual update be required? There are also questions about

using SCCM Collections. A separate collection might be required for each licensed application in order to have full control of deployment.

5.10 Comparison

The two models presented have notable differences. Software Center enables very basic, bare-bones functionality that gets the job done, but does not meet demands that were set for application distribution and approval. Using Software Center for application distribution would essentially mean return to the old model that was used before SCCM R2 update. This would be a largely email based model where approvals are done by either replying to notification emails or using a Sharepoint site. The much talked application store like experience would not be achieved this way and no see-through approval workflow would exist. Only clear advantage is that no additional solutions would be needed and “clean” Microsoft-environment would be preserved. There would either not be any additional direct costs. In a nutshell, continuing with Software Center alone is something that can be done and lived with, but will not advance software distribution in any way and therefore is not a desirable solution. We can do better and there are options.

Using a ServiceNow and Shopping based application store and workflows will be huge advancement compared with Software Center. Technical advantages like customizable application store and a see through approval process have already been talked about. The really big deal here is how it will answer the core problem: lack of understanding about the cost of software. So far software prices have been something that only a handful of people know about. It is not that they are a secret, but there has not been a way to present prices for users, managers and owners. Currently used Software Center simply does not allow it. The result can be guessed. Cost awareness cannot be achieved when there is no price information displayed anywhere. Applications have been requested and approved without any knowledge about how much money was just spent. This has resulted in high software costs and large numbers of unnecessary software installations. Displaying price information for every application is not an issue in the Shopping application store. It should result in considerably better awareness about software costs, which will lead to a smaller number of requests of expensive applications

(when something is expensive, people will think about it more closely before requesting) and managers will be more careful in approving requests when they actually see how using application will affect their budget. I believe that this alone will more than cover the price of implementing the ServiceNow based application store.

5.11 Option: Automys Software Deployment – ConfigMgr

Automys Software Deployment – ConfigMgr is similar, alternative solution to 1E's Shopping. Base idea is the same, integrate ServiceNow into the underlying SCCM architecture to create system where applications can be browsed, installed and requested via a simple to use store interface, which automatically creates tickets which can be used for processing requests (Stahl N. 2015a). Automy Inc is small software company founded in 2014 and is headquartered in Tampa, FL USA. The main products are automation solutions designed for companies to make daily processes easier. Software Deployment – ConfigMgr is a ServiceNow application sold via ServiceNow Application Store. As mentioned earlier, ServiceNow is a platform that can be customized and expanded via applications available through the platforms application store.

Automys solution uses existing ServiceNow service catalog as an application store, instead of its own interface like in 1E's Shopping (Automys. 2015). This catalog can be used to browse, request and install applications. It features very similar customization as Shopping, with capability to show price information, description, screenshots etc. This is an important difference when with compared to Shopping, since this means that the product sold here is the integration between ServiceNow and SCCM. SCCM application catalog is automatically discovered and imported into ServiceNow catalog, this import can be customized with filters for example (Stahl N. 2015b). No SCCM changes are needed besides access to a service account. Requests and approvals are done using ServiceNow's workflow automation, very similar to 1E's Shopping except that in Automys solution request records are used instead of incidents. Like incidents, request records provide workflow and tools for a multistage approval process that can be accessed by all parties to

provide information who are participating and how. After approval is completed, installation via SCCM is started.

Compared with 1E Shopping, Automys solution has some advantages but also disadvantages. Main advantage is that it is a lighter and simpler solution, with no additional third party application store and data base like in Shopping. Because of this, it is also likely considerable cheaper and somewhat simpler to implement, as it requires no changes to underlying SCCM configuration. However, what is gained in being simple, is losed in functionality. ServiceNow Service catalog used is not as easy to use, good looking and customizable as Shopping's Store. The interface is more like ticketing system GUI than a real consumer oriented store. Another and, in my opinion, decisive difference is that Automys is very young and new company with no references or comprehensive support resources. This means that "selling" this solution to decision making people will be difficult. It is easy to think Automys is too small to deliver such a product to large enterprise like this. And there also might some truth to this. A capable and stable provider is needed as it brings certain stability and continuity.

6 WILD IDEAS

During several meetings that were held when writing this thesis, some fresh and wild ideas about application deployment were brought up. Being open minded was encouraged and new ideas were welcome. These ideas were never refined to level where technical details would be considered, instead they were more like concepts of the approval processes itself. Who is going to process requests, who decides, how will installation happen? In the end all of these ideas were scrapped since it was deduced that it would not be possible to execute them in real life. I am going to go through a few of these in short.

6.1 Let's approve everything

Since IT costs will be charged from every department separately instead of one, companywide IT budget starting in 2016, one approach could be just to approve everything. If you use it, you pay it. In practice this means that every

department will be responsible for their own IT costs and it could be assumed that if a request for licensed application was made, it would already be approved inside the department from where the request was made. Why exercise “outside control” about things that are not ours to control in the first place? Application licenses are no longer in IT budget so why would IT care about the license costs of each department? In this concept IT would not take any part in decision making or approval process, instead it would only act as the “logistics” who will deliver the requested application in working condition.

In practice there would no longer be a split to licensed and freeware applications in the application catalog. Everything would be “freeware” and could be installed immediately without making a request. No approval process would exist. Cost changing would happen based on installed applications in SCCM database. The application owner would still continue to have the responsibility of maintaining licenses, they would just no longer be part of the approval process. In essence this is an “every man for themselves” –model. There is no control from “upper level”, nobody is babysitting people and check what they want to install. If you screw up, then you face the consequences. If your department exceeds its budget because of high application license costs, then it is up to you explain why it happened and make sure it won’t happen again. Basically, this does not differ in any way from situation where budget is exceeded because electricity costs are too high or too many personnel are employed and salary costs are too high.

In the end, this model was considered as too radical and daring. As has been stated before, one core issue is that people lack the knowledge about software costs and have a tendency to install applications even if they do not have a real need for them. If everybody would suddenly be able to install anything available in the application catalog, there would be large risk of the number of application installations getting out of hand before anyone would notice and take action. In organization this big, the amount of money in question could be huge. Model like this would require people who have understanding about software costs and ability to act responsibly.

6.2 Full IT control

Often the varying performance of application owners and managers in processing application request sparked an idea about doing everything from start to finish by ourselves. Most application owners, for example, are not IT people. Application ownership is something that they do as a secondary job and in many cases it is responsibility that they have not taken because they wanted to, but because according to company practice the original requestor of application will be named as application owner. There are some cases where it can clearly be seen that the application owner is not very motivated to handle his task. If we want to be absolutely sure that application requests are processed in the best possible manner, then why not give the responsibility to the people who are professionals in application management and IT in general?

This model can be applied to any technical implementation, since it does not take any stance on how application deployment and approval is made in technical sense. There are two parties taking part in processing request, application management and users manager. When request is made, it would be routed to users manager (read from AD), who would give decision (approve/deny) and then the request will be forwarded to application management, where license check and final approval are done. The application owner would not be involved in this. This is the “let’s do it ourselves” –model. It is based on the idea that people doing application management as their main responsibility have interest in making sure that things are being done by the best possible way.

One additional option would be to continue use Software Center as application catalog but remove all licensed software from Software Center completely, and keep it only as a distribution channel for freeware applications. This could reduce the number of unnecessary application requests since licensed application would be “hidden” from sight. Applying for licensed applications would be done by sending email to application management, which would process it as described earlier.

The paint point in this model is available human resources in IT. Since the average daily number of application requests is considerable, processing all of them inside IT department would require new resources. It was widely agreed

that current amount of staff is not enough if this kind of work is to be done in IT. Some also thought that this not job for 3rd-level IT engineers, but something that should be in Service Desk. The final question is the position of the application owners. If the application owner is completely taken away from license checking, it can result in situation where the owner is longer aware where the application he/she manages is being used and for what. This conflicts with the basic idea of application ownership.

7 CONCLUSION

When making this thesis, it became clear that application management in enterprise is a complex and challenging subject that is also relatively young. Finding information and literature about it proved difficult. Availability of ready-made solutions was also surprisingly low, almost non-existent. At first this seems odd, surely all companies are wrestling with the same issues and when there is a need, there are also solutions to answer that need. Unfortunately enterprise world is slow moving, change takes time, lots of time. Things like an enterprise application store and BYOD for example are relatively new and present challenges that are unseen in enterprise environment. The direction is slowly but steadily towards more diverse environment where everybody no longer uses the same devices running the same software. Instead people will bring their own favorite devices to work and expect everything to work as usual. There is a large difference between enterprise world and consumer space. What has been daily reality in consumer space, is only now starting slowly to appear enterprise space. Application deployment is a great example of this. During the making of this thesis, I had change to ask some people about application distribution in some large IT companies and answers were surprising. Every one of these companies are either using an email solution were in order to install software, user had to directly contact support and have someone install the needed application, or there was a Software Center type, very basic application store in use with an email based approval process. There were also combinations of these two ways in use. In these cases only freeware applications were distributed via application store and licensed had be requested via direct contact to support or application management.

For a long time, application distribution has been handled via local installations and email based approval processes like described above, but those are no longer viable ways of doing things. In small companies, they might be a viable way, but not in large enterprises. The application store like experience that is mentioned many times in this thesis, is something that is the norm in consumer space. Something that everybody uses and no platform can survive without. It is easy to think that concept used so widely in consumer space would also be highly utilized in enterprises as it has very clear and considerable advantages, but that unfortunately is not the case. Like in our client company, it seems that most companies are only now starting to wake up to the situation and begin thinking about what could be done to improve it. Likewise, it seems that software companies have only recently started to look into developing solutions to application deployments. Some companies like Microsoft do not even seem to realize the need for more advanced application deployment methods or they do not see it as profitable business. Forthcoming new SCCM for example does almost nothing to improve application discovery, approval and deployment, which is surprising and also lost potential for Microsoft. This of course makes a way for smaller and more agile competitors like ServiceNow and 1E.

7.1 Proposal

As a result of this thesis, I see two realistic ways to handle application deployment in the future. It can be done via the use MS SCCM provided tools like Software Center or using ServiceNow combined with Shopping. For me, one these models represents advancement and other does not. Using Software Center it is possible to cobble together an email based approval process that will suffice in doing the very basic stuff, but nothing more. It will not provide solutions of any kind to various issues presented in this thesis nor shall we be able to achieve the much talked application store like experience and proper approval workflow. In a nutshell, using Software Center is a return to the same old ways of doing things. If the aim is to truly take a step forward and build something that is modern and long lasting, the Software Center will not be the way to do it.

My recommendation is to take the leap and go with ServiceNow and 1E Shopping. Why do something the half way when you can go all the way to the end? ServiceNow and Shopping answer most issues and challenges in application deployment.

- A modern, fully featured application store with extensive customizability.
- An easy to use, modern web-based interface.
- Capability to display price information within the application store.
- Automated, see-through approval process with full SCCM integration.
- One single interface for application store and ticketing system.
- Possibility for considerable cost saving through better awareness about software costs.

ServiceNow and Shopping have huge benefits when compared with Software Center. It comes with some added complexity and costs, but I believe that in the long run a return of investment will be seen as well as increased user satisfaction. It will also improve the decision making of managers and application owners who benefit from having proper approval workflow which provides them all necessary information like software cost. No longer will decisions be made based on inadequate information.

7.2 The human factor

Finally, it is very important to notice that humans play a key role in this matter. The root cause of issues in application approval is not technical but a human issue. In a perfect world, no approval process would be needed since everybody would take responsibility for their own actions and think about the bigger picture. A lot of the issues exist because of people have habit of installing software on very lightweight basics and not thinking about the resulting costs. It is unfortunate, but reality. This leads us to the realization that no matter what kind of system is built, it will not produce the wanted results if people are not utilizing it fully. Building a working application approval process not only requires a technically competent solution, but also people who are committed on making it all work.

When a new application request and approval process is taken into production, it is very important that especially key personnel like managers and application owners are properly informed and educated to utilize it. Users must be made aware of the new system and encouraged to utilize it. It is easy to think that all this not very relevant and focus just on building technically competent solution, but in fact the human side is very likely the most challenging aspect of the whole matter. If this fails, the full potential will not be achieved.

7.3 Final words

The subject of the thesis was new and challenging for me and I would not have been able to complete this alone. Many people supported me during the writing of this thesis and their support is highly valued. Researching the matter and mapping for possible solutions has taught me a lot about application management and challenges that it poses. It is a surprisingly complex matter that can easily appear as something simple. Many challenges are human related instead of being purely technical issues. I hope and believe that the results can be made use of when implementing a better application distribution and approval process in future.

8 LIST OF REFERENCES

1E Limited. 2015a. Shopping and ServiceNow integration technical presentation. Available: http://www.1e.com/blog/all_resources/shopping-servicenow-integration-technical-presentation/ [Accessed: 13 December 2015].

1E Limited. 2015b. Shopping and ServiceNow integration. Available: http://www.1e.com/blog/all_resources/shopping-servicenow-integration/ [Accessed: 13 December 2015].

1E Limited. 2015c. Shopping Product Sheet. Available: http://www.1e.com/downloads/documents/1E_Shopping_DataSheet.pdf [Accessed: 13 December 2015].

Advisera. 2012. 20000 Academy. ITIL Application Management Function – Custodian of application knowledge. Available: http://advisera.com/20000academy/knowledgebase/itil-application-management-lifecycle-within-service-lifecycle/ [Accessed: 5 October 2015].

Mac Neela A. 2013. What makes a great application owner? Available: http://blogs.forrester.com/alan_mac_neela/13-03-01-what_makes_a_great_application_owner [Accessed: 28 November 2015].

Automys. 2015. Self-Service App Store Integration: ServiceNow + Configuration Manager. Available: https://www.youtube.com/watch?v=Twp_-Rme_4k [Accessed: 10 December 2015].

Godfrey I. 2015. Shopping requests approval – Thumbs up for 1E's enterprise app store. Available: http://www.1e.com/blogs/2015/12/11/shopping-requests-approval-thumbs-up-for-1es-enterprise-app-store/ [Accessed: 15 December 2015].

Microsoft Corporation 2016a. Components of System Center 2012 R2. Available: https://www.microsoft.com/en-us/server-cloud/products/system-center-2012-r2/Components.aspx [Accessed: 2 October 2015].

Microsoft Corporation 2016b. Orchestrator Capabilities. Available: https://technet.microsoft.com/en-us/library/hh420338.aspx [Accessed: 2 October 2015].

Microsoft Corporation 2016c. System Center 2016. Available: https://www.microsoft.com/fi-fi/server-cloud/products/system-center-2016/ [Accessed: 18 November 2015].

Peterson N. 2014. Configuration Manager Application Approval Engine. Available: http://blogs.technet.com/b/neilp/archive/2014/09/10/cmaae.aspx [Accessed: 14 September 2015].

Peterson N. 2012. System Center 2012 Configuration Application Request / Approval – Deep Dive and Automation. Available: http://blogs.technet.com/b/neilp/archive/2012/09/18/system-center-2012-configuration-application-approval-deep-dive-and-automation-part-1.aspx [Accessed: 14 September 2015].

ServiceNow. 2015. Microsoft SCCM Integration 2012. Available: http://wiki.servicenow.com/index.php?title=Microsoft_SCCM_Integration_2012#gsc.tab=0 [Accessed: 18 December 2015].

Stahl N. 2015a. ServiceNow App: Software Deployment – ConfigMgr. Available: https://automys.com/products/servicenow/software-deployment-configuration-manager-sccm [Accessed: 17 December 2015].

Stahl N. 2015b. Self-Service Software Deployment - ServiceNow and System Center. Available: https://automys.com/library/asset/self-service-software-deployment-servicenow-system-center [Accessed: 17 December 2015].

Tulloch M. 2013. Introducing Microsoft System Center 2012 R2 Technical Overview. Available: https://mva.microsoft.com/ebooks [Accessed: 10 January 2016].

Wikipedia. 2015a. ServiceNow. Available: https://en.wikipedia.org/wiki/ServiceNow [Accessed: 20 November 2015].

Wikipedia. 2015b. Bring your own device. Available: https://en.wikipedia.org/wiki/Bring_your_own_device [Accessed: 1 October 2015].

Wikipedia 2016a. Computer. Available: https://en.wikipedia.org/wiki/Computer [Accessed: 5 September 2015].

Wikipedia 2016b. Operating system. Available: https://en.wikipedia.org/wiki/Operating_system [Accessed: 5 September 2015].

Wikipedia 2016c. Application software. Available: https://en.wikipedia.org/wiki/Application_software [Accessed: 5 September 2015].

Wikipedia 2016d. App store. Available: https://en.wikipedia.org/wiki/App_store [Accessed: 5 September 2015].

Wikipedia 2016e. Software License. Available: https://en.wikipedia.org/wiki/Software_license [Accessed: 7 September 2015].

Wikipedia 2016f. Application lifecycle management. https://en.wikipedia.org/wiki/Application_lifecycle_management [Accessed: 12 September 2015].

Wikipedia 2016g. Microsoft Servers. Available: https://en.wikipedia.org/wiki/Microsoft_Servers#Microsoft_System_Center [Accessed: 20 September 2015].

Wikipedia 2016h. System Center Configuration Manager. Available: https://en.wikipedia.org/wiki/System_Center_Configuration_Manager [Accessed: 12 January 2016].